Com & Link

asm86all.com c86lnk.a86

```
$ set def [frank.mpm86.asm86]
$ plm86 mainp.plm debug optimize(2) 'p1' xref
$ plm86 cmac1.plm debug optimize(2) 'p1'
$ plm86 cmac2.plm debug optimize(2) 'p1'
$ plm86 cmac3.plm debug optimize(2) 'p1'
$ plm86 cmac4.plm debug optimize(2) 'p1'
$ plm86 cmac5.plm debug optimize(2) 'p1'
$ plm86 mnem1.plm debug optimize(2) 'p1'
$ plm86 mnem2.plm debug optimize(2) 'p1'
$ plm86 mnem3.plm debug optimize(2) 'p1'
$ plm86 mnem4.plm debug optimize(2) 'p1'
$ plm86 symb.plm debug optimize(2) 'p1' xref
$ plm86 io.plm debug optimize(2) 'p1' xref
$ plm86 subr1.plm debug optimize(2) 'p1' xref
$ plm86 subr2.plm debug optimize(2) 'p1' xref
$ plm86 files.plm debug optimize(2) 'p1' xref
$ plm86 scan.plm debug optimize(2) 'p1' xref
$ plm86 print.plm debug optimize(2) 'p1' xref
$ plm86 predef.plm debug optimize(2) 'p1' xref
$ plm86 ermod.plm debug optimize(2) 'p1' xref
$ plm86 text.plm debug optimize(2) 'p1' xref
$ plm86 outp.plm debug optimize(2) 'p1' xref
$ plm86 expr.plm debug optimize(2) 'p1' xref
$ plm86 brexpr.plm debug optimize(2) 'p1' xref
$ plm86 pseud1.plm debug optimize(2) 'p1' xref
$ plm86 pseud2.plm debug optimize(2) 'p1' xref
$ plm86 cmsubr.plm debug optimize(2) 'p1' xref
$ plm86 instr.plm debug optimize(2) 'p1' xref
$ plm86 dline.plm debug optimize(2) 'p1' xref
$ plm86 global.plm debug optimize(2) 'p1' xref
$ plm86 cm.plm debug optimize(2) 'p1' xref
$ plm86 cm2.plm debug optimize(2) 'p1' xref
$ asm86 c86lnk.asm debug
$ link86 cmac1.obj,cmac2.obj,cmac3.obj,cmac4.obj,cmac5.obj to f11.mod
$ link86 mnem1.obj,mnem2.obj,mnem3.obj,mnem4.obj,symb.obj to f12.mod
$ link86 io.obj,subr1.obj,subr2.obj,files.obj,scan.obj to f13.mod
$ link86 print.obj,predef.obj,ermod.obj,text.obj,outp.obj to f14.mod
$ link86 expr.obj,brexpr.obj,pseud1.obj,pseud2.obj,cmsubr.obj to f15.mod
$ link86 instr.obj,dline.obj,global.obj,cm.obj,cm2.obj to f16.mod
$ link86 f11.mod,f12.mod,f13.mod to f21.mod
$ link86 f14.mod,f15.mod,f16.mod to f22.mod
$ link86 c86lnk.obj,mainp.obj,f21.mod,f22.mod,plm86.lib to asm86.mod
$ loc86 asm86.mod nopublics ad(sm(code(0))) od(sm(code,const,stack))
$ h86 asm86
! on micro under cp/m or mp/m
!
! determine BBB and MMM from asm86.mp2 file
! BBB = start of const segment / 16
! MMM = (start of memory segment - start of const segment)/ 16 + 100h
!          (4K for sysmbol table)
!
! gencmd asm86 data[bBBB,mMMM,xfff]
!    which turns out to be:
! gencmd asm86 data[4ad, m44e, xfff]
```

CCP/M-86 2.0 V.3
COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # CC-104

```
ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE C86LNK
OBJECT MODULE PLACED IN :F0:C86LNK.OBJ
ASSEMBLER INVOKED BY:  :F0: C86LNK.ASM DEBUG

LOC  OBJ                  LINE     SOURCE

                             1     ;
                             2             extrn   asm86:near
                             3     cgroup  group   code
                             4     dgroup  group   const,data,stack,memory
                             5
                             6             assume  cs:cgroup,ds:dgroup
                             7
----                         8     data    segment public  'DATA'
----                         9     data    ends
                            10
                            11     ;
----                        12     stack   segment word stack 'STACK'
0000                        13     stack_base      label   byte
----                        14     stack   ends
                            15     ;
                            16
----                        17     memory  segment memory  'MEMORY'
----                        18     memory  ends
                            19
----                        20     const   segment public  'CONST'
                            21
                            22             public  fcb,fcb16,tbuff,endbuf
                            23
0006                        24             org     6
  0006                      25     endbuf  equ     $
005C                        26             org     5ch
  005C                      27     fcb     equ     $
006C                        28             org     6ch
  006C                      29     fcb16   equ     $
0080                        30             org     80h
  0080                      31     tbuff   equ     $
0100                        32             org     100h
                            33
----                        34     const   ends
                            35
----                        36     code    segment public  'CODE'
                            37
                            38             public  mon1,mon2
                            39
0000 8CD8                   40     start:  mov     ax,ds
0002 9C                     41             pushf
0003 5B                     42             pop     bx
0004 FA                     43             cli
0005 8ED0                   44             mov     ss,ax
0007 8D260000         R     45             lea     sp,stack_base
000B 53                     46             push    bx
000C 9D                     47             popf
000D E90000           E     48             jmp     asm86
                            49
0010 20434F50595249         50     copyright       db      ' COPYRIGHT (C) DIGITAL RESEARCH, 1981 '
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
LOC  OBJ                  LINE     SOURCE

     47485420284329
     20444947495441
     4C2052455345 41
     5243482C203139
     383120
                           51
                           52              public  patch
0036                       53       patch:
0036 90                    54               db      90h,90h,90h,90h,90h,90h,90h,90h
0037 90
0038 90
0039 90
003A 90
003B 90
003C 90
003D 90
003E 90                    55               db      90h,90h,90h,90h,90h,90h,90h,90h
003F 90
0040 90
0041 90
0042 90
0043 90
0044 90
0045 90
0046 90                    56               db      90h,90h,90h,90h,90h,90h,90h,90h
0047 90
0048 90
0049 90
004A 90
004B 90
004C 90
004D 90
004E 90                    57               db      90h,90h,90h,90h,90h,90h,90h,90h
004F 90
0050 90
0051 90
0052 90
0053 90
0054 90
0055 90
0056 90                    58               db      90h,90h,90h,90h,90h,90h,90h,90h
0057 90
0058 90
0059 90
005A 90
005B 90
005C 90
005D 90
005E 90                    59               db      90h,90h,90h,90h,90h,90h,90h,90h
005F 90
0060 90
0061 90
0062 90
0063 90
0064 90
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
LOC  OBJ                  LINE     SOURCE

0065 90
0066 90                     60             db      90h,90h,90h,90h,90h,90h,90h,90h
0067 90
0068 90
0069 90
006A 90
006B 90
006C 90
006D 90
006E 90                     61             db      90h,90h,90h,90h,90h,90h,90h,90h
006F 90
0070 90
0071 90
0072 90
0073 90
0074 90
0075 90
                            62
0076 2031302F30312F         63     date    db      ' 10/01/81 '
     383120
                            64
0080                        65     bdos:
0080 58                     66             pop     ax      ; return address
0081 5A                     67             pop     dx
0082 59                     68             pop     cx
0083 50                     69             push    ax
0084 CDE0                   70             int     224
0086 C3                     71             ret
                            72
  0080                      73     mon1     equ    bdos
  0080                      74     mon2     equ    bdos
----                        75     code    ends
                            76
                            77     end

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____